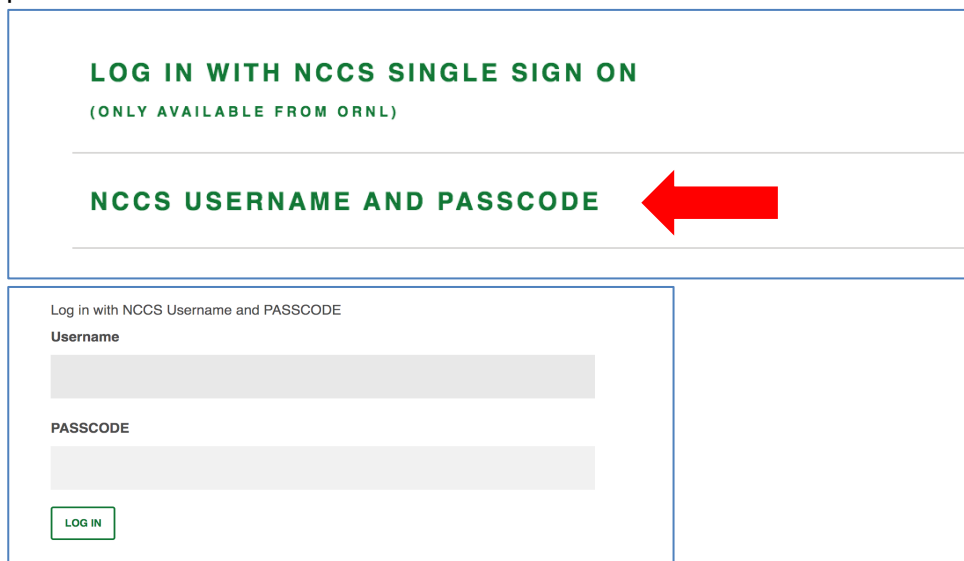


DAW Hands-On Session: Manual

Gateway Tutorial

1. In any browser window, enter the following URL: <https://jupyterhub-gen119blazingsql.apps.marble.ccs.ornl.gov/>
2. Choose the option to log in with NCCS user ID and password, and enter your NCCS user ID that was assigned with the OLCF account allocation and the pin+token as your password:



LOG IN WITH NCCS SINGLE SIGN ON
(ONLY AVAILABLE FROM ORNL)

NCCS USERNAME AND PASSCODE

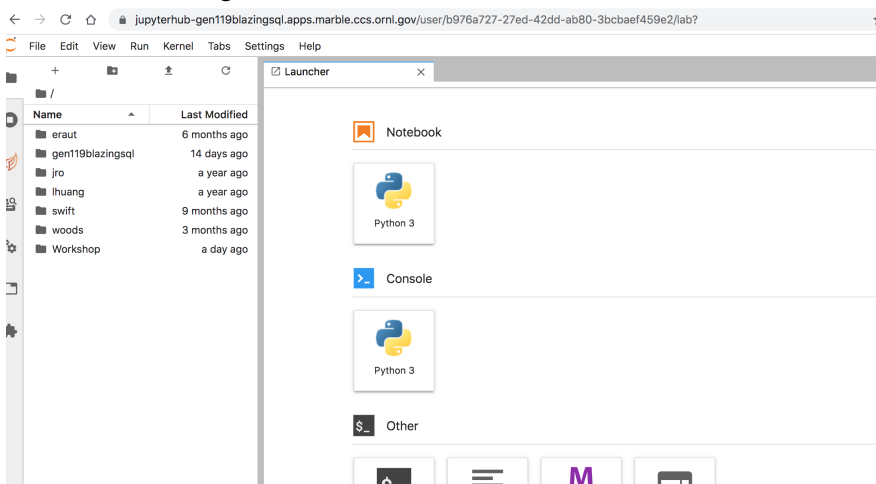
Log in with NCCS Username and PASSCODE

Username

PASSCODE

LOG IN

3. After entering this information, you should see the JupyterHub interface loading up and then the following screen:



4. On the left panel, double click on the Workshop folder to enter. Right click and choose “New Folder,” then change the folder name to something personalized for you (like your name).
5. Right click on the notebook called “gateway-example.ipynb” and choose “Copy.”
6. Navigate into your directory, and paste the notebook copy into your directory.
7. Double-click on the notebook in your directory. It should populate your screen:

```

[1]: import dask
    from dask.distributed import Client
    from dask_gateway import Gateway

[2]: gateway = Gateway(
    address="http://dask-gateway:8000",
    public_address="proxy/dask-gateway:8000"
)

[3]: options = gateway.cluster_options()

[4]: options
VBox(children=(HTML(value='<h2>Cluster Options</h2>'), GridBox(children=(HTML(value='<p style='font-weight: bo...

[5]: cluster = gateway.new_cluster(options)

[6]: cluster
VBox(children=(HTML(value='<h2>GatewayCluster</h2>'), HBox(children=(HTML(value='<n<div>\n<style scoped>\n

[7]: %jobs

```

JOBID	USER	STAT	SLOTS	QUEUE	START_TIME	FINISH_TIME	JOB_NAME
405128	gen119blaz	RUN	43	batch-hm	Oct 12 10:45	Oct 12 12:45	dask-gateway
405130	gen119blaz	RUN	43	batch-hm	Oct 12 10:46	Oct 12 12:46	dask-gateway

- Starting at the top, click on each box marked with a number on square brackets, from [1] through [4]. Press “shift” and “enter” at the same time to perform the set of commands that are pre-entered there. You may see an asterisk instead of the number briefly while the command is being performed. After this, the line number will appear in a lighter font. After you do this for line [4], you should see the following appear in response to the command asking for options:

```
[ 4]: options
```

Cluster Options

Number of worker nodes:

Job duration:

OLCF account:

Worker queue:

Scheduler queue:

BSUB alloc_flags:

This displays the options associated with your batch job submission to Summit through this JupyterHub interface. Here we are asking for 1 worker, for 2 hours, using the GEN119BlazingSQL account, the high-memory nodes which are in the batch-hm queue partition, and we are asking for use of the NVMe Burst Buffer, an SSD drive that comes with every compute node when this “alloc_flags” option is requested. At any point, if you want to stop the notebook from executing a command, you can hit the stop button in the top toolbar (dark square).

- The next command, `<<cluster = gateway.new_cluster(options)>>` contacts Marble on Slate so Marble can submit a job on Summit for you. This may take a few minutes to run.
- After the cluster command finishes, you now have a node to use as a scheduler. Now you need to request a Worker node. Press shift + enter on the `<<cluster>>` command. It should show the following:

GatewayCluster

Workers 1
Cores 1
Memory 2.15 GB

▼ Manual Scaling

Workers Scale

► Adaptive Scaling

Name: b5aab29b2e1147359ad9261f8abba239

Dashboard: <proxy/dask-gateway:8000/clusters/b5aab29b2e1147359ad9261f8abba239/status>

This is an interactive window. In the Manual Scaling area, increase the number in the Workers field to 1, and press the “Scale” button. Then wait for another Summit node to be allocated to you. The Workers list on the left hand side will automatically update to 1 when you have your Worker node. While you wait, you can go ahead and enter on the `<<!jobs>>` command. This is a Summit job scheduler command that returns the current running and pending jobs in this queue.

11. After you get your worker node, evaluate the `<<client = Client(cluster)>>` command. Don't worry about the pink warning message that appears. It is warning about a mismatch in versions that is not important. After this, enter the next command, `<<client>>`. It will return a summary of your client setup with your Worker:

Client

Scheduler: <gateway://dask-gateway:8000/b5aab29b2e1147359ad9261f8abba239>
Dashboard: <proxy/dask-gateway:8000/clusters/b5aab29b2e1147359ad9261f8abba239/status>

Cluster

Workers: 1
Cores: 1
Memory: 2.15 GB

12. Now, enter in the next two command regions, and you should see a greeting from your worker node's hostname.

```
import socket
```

```
print(client.submit(lambda: "Hello from "+socket.gethostname()).result())
```

```
Hello from c37n12
```

13. Now, let's try with two workers. Go back to the GatewayCluster window, and increase the number of workers to 2 in the Manual Scaling window. Press the Scale button and wait for a second Worker to be assigned.
14. Now go to the bottom of the page and enter in the final command. You should now see a greeting from both Workers:

```
[12]: futures = client.map(lambda x: "Hello from task {} on {}".format(x)+socket.gethostname(), range(2))
      for s in client.gather(futures):
      print(s)
```

```
Hello from task 0 on c37n12
Hello from task 1 on c37n13
```

15. Now look on the panel on the left-hand side. Click on the Dask icon:



Then on the search icon (magnifying glass). The rectangular boxes should turn orange. Clicking on any of these will open a new window in your dashboard showing data related to that aspect of your workflow. Each one of these windows can be arranged by dragging to create your customized dashboard, that lets you monitor your workflow while your tasks are being executed.

The screenshot displays the Dask Gateway dashboard interface. On the left, a code editor shows the following Python code:

```
[1]: import dask
from dask.distributed import Client
from dask_gateway import Gateway

[2]: gateway = Gateway(
    address="http://dask-gateway:8000",
    public_address="proxy/dask-gateway:8000"
)

[3]: options = gateway.cluster_options()

[4]: options
```

Below the code editor, the "Cluster Options" section is visible, showing input fields for "Number of worker" (set to 1), "nodes:" (empty), "Job" (set to 2:00), and "duration:" (empty).

On the right side of the dashboard, there are several panels:

- Dask Aggregate Time Per Action**: A bar chart showing time per action for various tasks. The "compute" task is highlighted in purple.
- Dask Workers**: A table showing the status of workers.

name	address	nthreads	cpu	memory	memory_lim
Total (2)		2	1.0 %	773 MiB	4 GiB
87320ec190b0	ts://10.134.21.1	1	0.0 %	387 MiB	2 GiB
a71bf7abe5764	ts://10.134.21.1	1	2.0 %	386 MiB	2 GiB

Below the workers table, there is a section for **Dask Bandwidth Workers** with a bar chart titled "Bandwidth by Worker" showing bandwidth usage for different workers.

Congratulations! You have finished the Gateway tutorial. To exit, choose File -> Shut Down.